Integrating and Ranking Interests From User Profiles*

Fabien Duchateau[1] and Lynda Hardman[1]

CWI, Science Park 123
1098 XG Amsterdam, The Netherlands
firstname.lastname@cwi.nl

Abstract. Many websites allow their users to personalize their profiles. As users subscribe to many personalization websites, such as social networks or search systems, each user owns different profiles, which are seldom compatible. Yet, there is a strong need for comparing the profiles of different users to discover shared interests, e.g., by integrating all user profiles into a global one. In this paper, we propose a novel method for integrating and ranking user interests from various profiles. Our approach relies on the identification of high-level concepts around which similar user interests are clustered. We compute the weight of each cluster with respect to the other ones, thus enabling the ranking of the most shared user interests between user profiles.

1 Introduction

Nowadays, a majority of Web users interacts daily with systems that store (some of) their preferences and interests. Indeed, these personalization websites have reached sufficient maturity to cover a large spectrum of applications such as e-commerce, search engines, social networks. Most of them already propose to their users recommendations or advertisements based on their profiles. In a similar fashion, they help users to find people for sharing common activities, dating or finding a roommate. Interoperability between these systems would benefit both users and information providers, by overcoming the issue of integrating user profiles [1]. We assume that user profiles are in the same language. Multilingual issues have been studied in [2] but they are out of scope of this paper.

In this context, a number of frameworks have been proposed to create and manage user profiles [3–5]. Other initiatives such as OpenID[1] provide users with a means of storage for their passwords and basic profile information (e.g., name, address). Companies are also interested in user profiles, for instance to build a pool of experts on a specific topic. In [6], authors integrate profiles from various users, in particular for human resources purposes. These approaches require specific user inputs. In [7], authors aim at making interoperable user models for

* The first author carried out this work during the tenure of an ERCIM "Alain Bensoussan" Fellowship Programme.

[1] http://openid.net

education. More specifically, they convert these user models from one system to another, mainly relying on manual integration. In addition, the work is restricted to two specific systems. Similarly to our approach, Li et al. also cluster labels/tags to discover user interests [8]. However, their approach requires user-annotated tags. Other initiatives like *Mypes*[2] already aggregate user profiles from different personalization websites, but they do not gather similar interests under the same concept, thus implying many redundancies in the generated tag cloud (e.g. "mountain" and "mountains").. In a similar fashion, *Google's Social Graph API*[3] enables to discover the relationship between people. However, it implies that there exists a public link between user profiles. For representing user profiles, several user models such as *Friend Of A Friend* [9] (FOAF), *General User Modelling Ontology* (GUMO) [10] or UserRDF [11] currently exist. While such modelling frameworks have been developed, many Web applications still include their own user models.

Many issues need to be addressed to ensure interoperability between Web applications, specifically for user profiles. For instance, one may require to integrate a user profile to a higher level of abstraction such as models like *GUMO*. In a similar fashion, users often own different profiles. Yet, if they need to create a new one, this process is manually performed from scratch. Thus, we believe there is a lack of integration approaches for user profiles to solve these problems. But more challenges need to be addressed at the application level too. Social networks and commercial websites intensively analyse their user profiles for recommendations. Due to the possible growth of these profiles, for which users can subscribe to thousands of groups for instance, how can we extract the most important interests for a given user? And when several users are involved, applications emphasize the comparison of their profiles to deduce their shared interests.

In this paper, we explore work which aims at tackling all these issues. Indeed, our work is intended to be useful both for end-users and for service providers. Owners of scattered user profiles would be able to merge them into a unified profile or could use information already stored in their profiles to automatically build a new one. On the other hand, service providers would benefit from our work for recommendation purposes by detecting and representing common user interests, even when these users do not share online connections. Thus, we first propose a method for integrating two profiles (either from the same user or from different ones). The idea is to cluster similar interests around a high-level concept. The discovering of these concepts and the matching of an interest towards a concept are performed using state-of-the-art matching tools. Then, we present a measure for assessing the importance of a cluster with respect to the other ones. It is based on the cluster relationships between interests and their concept to compute a weight. This weight enables the ranking of the most shared interests between the two profiles.

[2] http://mypes.groupme.org/mypes/
[3] http://code.google.com/apis/socialgraph/

The rest of this paper is divided as follows: Section 2 illustrates the problem in terms of a scenario. Section 3 describes our approach in detail. Finally, we conclude and outline several perspectives in Section 5.

2 Scenario and Approach

In this section, we first describe an example. Then, we present an overview of our approach.

2.1 A Running Example

Let us imagine two people looking for a medical job. Jane has a *Facebook* account and stores her bookmarks, including those about job searches, using *del.icio.us*. John also has a Facebook account for leisure and personal activities, but he stays tuned to professional networks with *LinkedIn* and his bookmarks are locally stored in his web browser. Figure 1 depicts this scenario. In parentheses, *I*, *G* and *P* respectively stand for interest, group or page that the user has subscribed to. We notice that both users share a common sport (*fishing*/*angling*), and that both enjoy other sports (*tennis*, *rock climbing*).

(a) Jane's profile (b) John's profile

Fig. 1. Two Examples of User Profiles from several Personalization Websites with (G): Group, (I): Interests and (P): page.

Note that we could have chosen to integrate profiles from the same user. For instance, by integrating *Facebook* and *Delicious* profiles from Jane, we would have noticed that she is strongly interested in finding a job. On the other hand, integrating John's profiles could lead to some suggestions for adding Facebook groups about *job searches*. In the rest of this paper, we focus on comparing profiles from both users and ranking their most common interests.

2.2 Overview of our Approach

Our goal is to find common interests between two user profiles. Directly matching individual interests using only terminological measures is not sufficient because if we match "angling" with "tennis", we could not discover the "sport" concept. We thus choose to include matching using concepts, since this allows identified terms to be related to (multiple) higher-level concepts. This also allows us to include pairs of interests in multiple clusters (in our example, "fishing" and "angling" are connected to cluster "fishing", but they are also connected to cluster "sport"). In addition, we believe that our approach should be generic. Thus, we assume that there are no semantics between interests included in the personalization websites. We also consider that user interests are not structured (e.g., with categories). For these reasons, we only apply terminological and linguistic similarity measures. While using both methods helps us satisfy our goal of integration and comparison of user profiles, we need some way of combining the information from both methods.

We propose a two-step approach, as shown by Figure 2. Users own profiles on one or more **personalization websites**. The first component is the **integrating component**, which extracts data from the user profiles. Then, these data are matched to *Wordnet* concepts, thus forming **clusters**. For comparing common interests from several user profiles, a next step is required to evaluate the weight of each cluster according to the number of interests it gathers. This is performed by the **ranking component** which outputs **ranked user interests**. The next two sections contain the details of each component.

3 Integrating Interests from User Profiles

This sections first describes the two steps for integrating user profiles: (i) extracting user interests and (ii) clustering them. The last part of this section is dedicated to discussion about these steps.

3.1 Extraction of User Profiles

During the first step, the integrating component extracts user interests from various personalization websites, e.g., *Facebook* or *LinkedIn*. These interests not only gather the list of activities and hobbies that the user filled in during profile creation, but also include the *groups* to which the user subscribes, the *pages* marked as "watch this page" and the *bookmarks* (s)he has added. All of these are

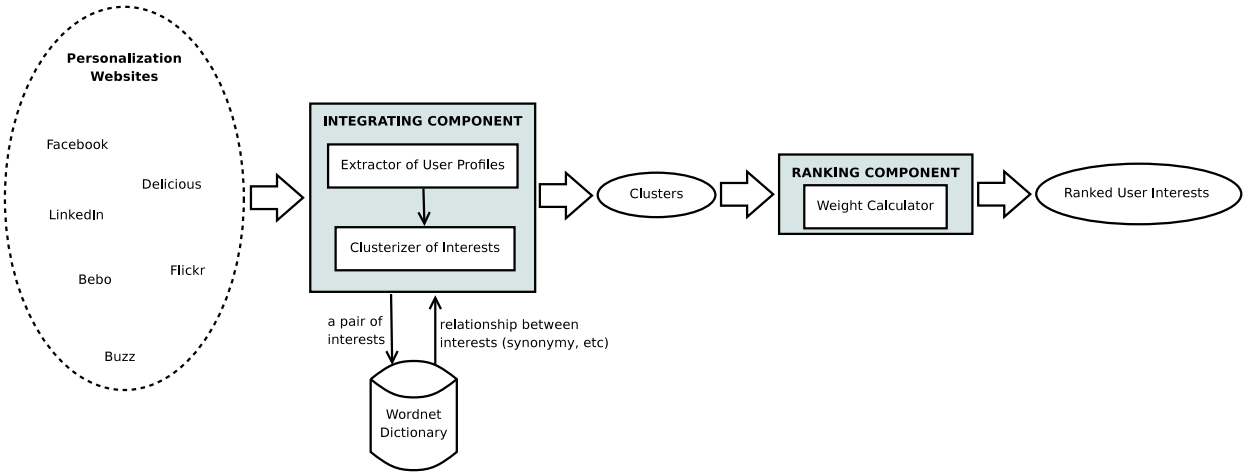

Fig. 2. Overview of our Approach

represented by a label (e.g., the name of a group, or the description of a bookmark). Currently, we are able to extract such information from profiles stored in *Facebook*, *LinkedIn* and *Del.icio.us*. The extractor uses common processes (tokenization, lemmatization and stemming [12]) to normalize each (label) interest for the matcher component. *In our running example, the interest medical professionals is normalized into two tokens, medical and profession. Similarly, the interest LastFM: listen to Internet music catalog is normalized into three tokens, namely listen, music, Internet and catalog. The token LastFM is discarded during this process because it does not have a Wordnet entry. Note that we could also create a cluster for these discarded tokens. In this case, the number of occurences for each token would eventually increase the importance of the interest.*

3.2 Clustering Interests around High-level Concepts

The second step performed by the integrating component is to discover concepts that gather several interests. In other words, we want to create clusters, each composed of a high-level concept (from Wordnet dictionary) and a list of interests that are related to this concept. To do so, we apply matching techniques, both linguistics and terminological [12], between all interests from any two profiles.

In our running example, this means that each of the 16 *interests from Jane's profile (including the two category interests tennis and work) would be matched to each of the* 14 *interests from John's profile.*

For linguistics (or semantics), we only use the *Wordnet*-based similarity measure from YAM [13], a matching tool. This measure enables the discovery of relationships such as hypernym, synonym or hyponym between two user interests. In other words, we check the set of ancestors in the Wordnet hierarchy for each interest. If we find a common ancestor between both interests, then this ancestor becomes a concept around which the two interests are clustered. Note that

in Wordnet, all words have a common ancestor (*entity*). To avoid discovering such abstract concepts when matching a pair of labels, we have constrained the hierarchy to 7 lower levels at most [14]. The Wordnet dictionary is also used in [15], in which the authors map the *Flickr* tags to *Wordnet* categories. However, these categories are high-level concepts (e.g., location, event, time) while our approach aims at finding the closest ancestor between two interests.

In our example, matching fishing and angling reveals that the former label is a hypernym of the latter. Consequently, the fishing concept is created. Similarly, matching work and profession enables the discovery of the job concept. Figure 3 depicts the four concepts that are created with the running example. Edges represented by a full line denote a linguistic relationship (discovered using Wordnet), and the number indicates how far, in terms of Wordnet relationships, the interest is from the concept (e.g., the fishing interest is related to the concept sport by two concepts, namely outdoor sport and sport).

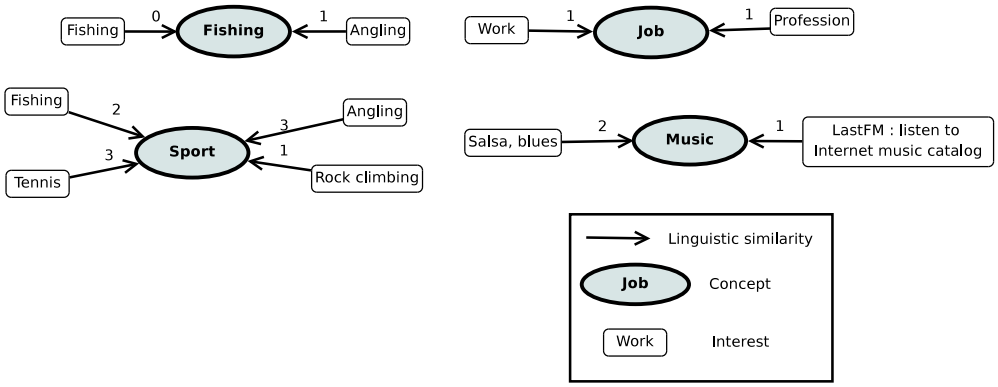

Fig. 3. Interests Linked to the Concepts using Linguistic Measures

Next, we apply terminological similarity measures (e.g., *Needle-Wunsche*, *Trigrams* [12]) to compare character strings of the interests. Namely, we aim at linking all interests that have not been linked either to a concept or to an interest linked to a concept. We have chosen to use COMA++ [16], a matching tool reputed to provide acceptable matching quality [17]. Indeed, this tool strongly promotes precision, thus avoiding the discovery of irrelevant correspondences. COMA++'s library contains 17 terminological measures that are aggregated into a global similarity value[4]. When the global similarity value computed for a pair of two labels is too low, COMA++ automatically discards this pair from the results list. This tool outputs a list of similar interests (according to their labels comparison) associated with a similarity value between 0 and 1 (1 denoting perfect similarity).

[4] Note that the list of normalized interests has been converted into a simple schema so that it can be processed by COMA++ and YAM.

In our example, COMA++ matches for instance USA jobs or Job search to the concept job with similarity values of 0.48 *and* 0.42 *respectively. As COMA++ mainly aggregates terminological measures, the interest USA jobs, whose character string is smaller than the one of Job search, has a higher similarity value with the concept job than Job search has. The tool also discards many candidate pairs, e.g., between USA jobs and angling whose similarity value is close to* 0.

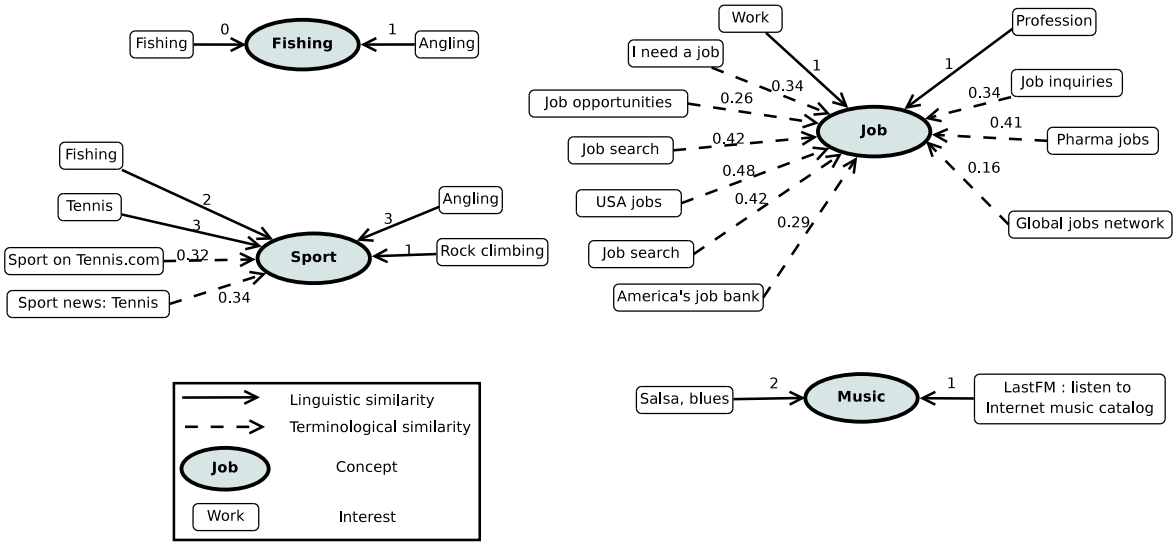

Fig. 4. Interests Linked to the Concepts using Terminological Measures

At the end of this step, similar interests have been clustered (or integrated) around the same concept, as shown by Figure 4. The similarity value computed by terminological measures (e.g., interest *Sport on Tennis.com* is terminologically similar to the *sport* concept with 0.32 confidence).

3.3 Discussion

Both interests and concepts are represented by URIs. This avoids confusion with similar labels. For example, the interest *job search* (group) is different from *job search* (bookmark) but *fishing* interest linked to *sport* concept is the same entity as *fishing* interest linked to the *fishing* concept.

In the *Wordnet* hierarchy, a concept may have different meanings. For instance, the concept *rock* can refer to a music genre, a stone, etc. If we try to discover a common concept between a fanatic of *progressive rock* and a geology fanatic, specifically for *chondrites* (a rock of meteoric origin containing chondrules), then we discover the common concept *rock* in the Wordnet hierarchy between the two user interests. To avoid this problem, we analyse the direct

hyponyms of all meanings for the discovered concept (*rock* in our example) in the *Wordnet* hierarchy. For the rock fanatic, its interest *progressive rock* appears under the following meaning "*rock 'n' roll, rock'n'roll, rock-and-roll, rock and roll, rock, rock music*". On the contrary, the geology fanatic has its interest *chondrites* under the meaning "*rock, stone*". Thus, we are able to disambiguate two interests that could have been clustered around the same concept due to its different meanings.

In our example, we only integrate two user profiles. However, we are able to integrate more than two profiles by means of two techniques: (i) incremental or (ii) holistic. The former is the fastest technique. Once we have integrated two profiles, we incrementally integrate another one by computing only terminological values, i.e., we directly match interests from the new profile with the concepts which have already been extracted between the initial user profiles. On the contrary, the holistic technique integrates each profile with all the others by applying both linguistic and terminological similarity measures. Thus, this technique allows to discover all concepts between all profiles, but to the detriment of execution time.

We notice a gap between the value distributions of the scores obtained either by terminological or by linguistic edges. Indeed, scores obtained for linguistic edges are higher than the terminologic ones. We believe that this clearly reflects the quality of the similarity measures which are used. On the other hand, terminological measures mainly return more similarities (although with a lower score) than the linguistic one, which compensates for the strictness of the latter.

Many user profiles may contain hundreds of interests, in particular groups and web pages. Thus, the number of extracted concepts can be very large, and this can lead to confusion or unusability. Therefore, we propose to rank the concepts (as well as user interests) using computed similarity values.

4 Ranking Interests from User Profiles

Ranking interests is required in some contexts, specifically to retrieve a particular one from a large collection. For instance, if a company is looking for a *Prolog* programmer, it is necessary to dissociate *Prolog* experts from users who have some *Prolog* tutorials in their bookmarks. Thus, we choose to rank concepts (and consequently interests underlying them) according to their importance in both profiles.

Each cluster that has been computed during a previous step can be seen as a connected component of a disconnected graph [18]. More formally, we have in our context:

- a set of vertices, composed of C, the set of concepts, and I, the set of interests
- E, a set of edges between a concept and an interest. Given $c \in C$ and $i \in I$, the edge e between c and i is noted e_{ci}. An edge either belongs to T, the set of terminological edges, or to L, the set of linguistic edges. Each edge e

has a weight noted $val(e)$. It corresponds to the similarity value computed in the previous step.

Our intuition is to compute, for each concept, its weight in the graph. This weight can be seen as the number and the quality of all interests that are linked to a concept. By quality of an interest, we mean the type of similarity measure used to link it to the concept (terminological or linguistic, the former being less trustable than the latter) and the associated similarity value (which assesses the confidence we have in the link). To do so, we first need to calculate the score of each interest related to a concept. Given a concept $c \in C$ and an interest $i \in I$ connected by an edge e_{ci}, we propose formula 1 to compute the score of an interest with respect to a concept.

$$score(c,i) = \begin{cases} val(e_{ci}) \text{ if } e_{ci} \in T \\ 1 \quad \text{if } e_{ci} \in L \text{ and } val(e_{ci}) = 0 \\ \frac{1}{val(e_{ci})} \quad \text{if } e_{ci} \in L \text{ and } val(e_{ci}) \neq 0 \end{cases} \tag{1}$$

Intuitively, the score of an interest is equal to its similarity value with the concept in case both are related via a terminological edge. On the other hand, with a linguistic edge, the score is inversely proportional to the number of intermediary (Wordnet) concepts between the interest and the concept. A specific case appears when the concept is identical to the interest and when they are linked by a linguistic edge. Although the number of intermediate concepts between them is null, the similarity between the concept and the interest is total. All computed scores are in the range $[0, 1]$.

*Let us compute several scores in our running example. The interest "USA jobs" is terminologically connected to the concept "job". Consequently, the score between them is equal to their similarity value (*0.48*). The concepts "tennis" and "sport" are linked by a linguistic edge, which indicates that they are separated by* 3 *concepts in the "Wordnet" hierarchy. Thus, the score between "tennis" and "sport" is equal to* $\frac{1}{3}$*. Finally, the interest "fishing" has no intermediate (Wordnet) concept with the concept "fishing", and its score is equal to* 1*.*

Now that we have a score between an interest and a concept, it is possible to compute the weight of each concept. The idea is to sum all scores between a concept and its interests because the more links a concept has, the more important it should be. In the following, $|I|$ denotes the number of distinct interests that have been linked to any concept. By distinct interest, we mean represented by different URIs, as explained in Section 3.3. Given a concept $c \in C$, we designate its related interests by a set $I_c = < i_1, i_2, ..., i_n >$ such that $\forall$ $i_k \in I_c$, $\exists$ $e_{ci_k} \in E$. Using these definitions, formula 2 computes the score of the concept c:

$$weight(c) = \frac{\sum_{k=1}^{n} score(c, i_k)}{|I|} \tag{2}$$

This formula returns values in the range $[0, 1]$. Indeed, the upper bound is reached when there is only one cluster whose edges all have the maximum value (1 for

a score). As for the lower bound, it tends to 0, although this value cannot be reached. As concepts are discovered by linguistic measures, there are at least two edges linking any concept. As the number of intermediate (Wordnet) concepts is a finite set, the scores of these linguistic edges tend towards 0 in the worst case. Thus, even if we imagine an infinite number of distinct interests linked to any concepts, the weight of any concept only tends towards 0.

What happens with the concepts from Figure 4 ? The number of distinct interests linked to one or more concepts is 19*. Let us compute the weight for the concept fishing. It has two linguistic relationships, and its score is therefore equal to* $\frac{1+\frac{1}{1}}{19} = 0.11$*. For the sport concept, we have both linguistic and terminological edges. We compute its weight using formula 2 to obtain a value equal to* $\frac{\frac{1}{1}+\frac{1}{2}+\frac{1}{3}+\frac{1}{3}+0.32+0.34}{19} = 0.15$*.*

At the end of this process, we have computed a score for each concept. Consequently, it is possible to rank them with the shared interests that are most represented at the top. These discovered concepts also form a "summary" between user profiles, since they are mainly higher level abstractions in the *Wordnet* dictionary (e.g., *rock climbing* and *tennis* w.r.t. the *sport* concept).

Following is the ranked list of common interests (with their scores in parentheses) shared by June and John's profiles in our running example:

1. *job (0.27)*
2. *sport (0.15)*
3. *fishing (0.11)*
4. *music (0.08)*

5 Conclusion

In this paper, we have focused on integrating different user profiles for ranking shared interests. As we desire a generic approach, we have only used linguistic and terminological similarity measures applied to interests. We have noticed that matching techniques are not perfect and can discover several irrelevant similarities between totally different interests. Specifically, this weak point mostly appears with terminological measures.

We see many perspectives to our work. At first, we aim at integrating profiles from other personalization websites. Namely, we need to identify these websites and build appropriate wrappers to extract user interests. Some web services or APIs are sometimes available to fulfill this goal. Concepts are currently browsed through *Wordnet* dictionary. However, other resources, such as *DBpedia*, could be useful to enrich or confirm the relationship between a user interest and a concept. For instance, the interest *salsa* only appears in *Wordnet* as a *spicy sauce*, thus no similar concept with music could be discovered whereas the *salsa* disambiguation page on Wikipedia lists more than 10 different meanings, including the *music style*. We will also explore the possibility to match user interests to concepts from models such as *GUMO*.

Another challenge would be the extraction of preferences from free texts (or unstructured texts) such as blogs or user reviews. A user who wrote a positive comment or gave a good rating to the *Lord of the Rings* books is likely to have interests in reading other *Tolkien* and/or fantasy books.

An improvement could be reached for integrating several user profiles. The *incremental* technique suffers from a possible missing of several concepts while the *holistic* one is time consuming. An idea to tackle these issues is a hybrid approach, which would apply the costly linguistic similarity measure only between interests that cannot be matched to an existing concept.

A last perspective deals with **user behaviours**. Returning to our running example, it is likely that our unemployed users regularly visit job search websites. Once they have found a job, they will probably not consult these websites for a while. Thus, frequency of visited websites is one of the measures that could help us to update most shared interests over time. Another example would be the discovery of another shared interest about *medical*. Although John joined a few groups related to medical and health on *LinkedIn*, it seems that Jane typically fills in forms with similar keywords to find jobs in her domain. Recording these keywords on job search websites would enable to detect that both users work in the same field. Further, we could also take into account the fact that different personalization websites are exploited in different contexts, e.g. Facebook is mainly used for contacting people, while LinkedIn is more used for professional purposes.

References

1. Aroyo, L., Dolog, P., Houben, G.J., Kravcik, M., Naeve, A., Nilsson, M., Wild, F.: Interoperability in personalized adaptive learning. Educational Technology & Society **9**(2) (2006)
2. Schultz, T., Kirchhoff, K.: Multilingual Speech Processing. Academic Press (2006)
3. Palmisano, I., Redavid, D., Iannone, L., Semeraro, G., Degemmis, M., Lops, P., Licchelli, O.: A rdf-based framework for user profile creation and management. In: SWAP. (2005)
4. Abel, F., Heckmann, D., Herder, E., Hidders, J., Houben, G.J., Krause, D., Leonardi, E., van der Slujis, K.: A framework for flexible user profile mashups. In: Proceedings of International Workshop on Adaptation and Personalization for Web 2.0 (AP-WEB 2.0 2009). CEUR Workshop Proceedings, Trento, Italy, CEUR, Tilburg, Aachen (June 22 2009) 1–10
5. Korth, A., Plumbaum, T.: A framework for ubiquitous user modeling. In: IEEE International Conference on Information Reuse and Integration (IRI2007), IEEE Computer Society Press (2007)
6. Vandermeulen, B., Duflou, J.R., Moor, B.D.: The role of user profiles in vector-based information retrieval. In: IKE. (2003) 668–669
7. Stewart, C., Cristea, A., Celik, I., Ashman, H.: Interoperability between AEH user models. In: APS '06: Proceedings of the joint international workshop on Adaptivity, personalization & the semantic web, New York, NY, USA, ACM (2006) 21–30
8. Li, X., Guo, L., Zhao, Y.E.: Tag-based social interest discovery. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 675–684

9. Brickley, D., Miller, L.: Foaf vocabulary specification 0.97 (January 2010)
10. Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., von Wilamowitz-Moellendorff, M.: Gumo - the general user model ontology. In: User Modeling. (2005) 428–432
11. Abel, F., Henze, N., Krause, D., Plappert, D.: User modeling and user profile exchange for semantic web applications. In: LWA. (2008) 4–9
12. Euzenat, J., Shvaiko, P.: Ontology matching. Springer-Verlag, Heidelberg (DE) (2007)
13. Duchateau, F., Coletta, R., Bellahsene, Z., Miller, R.J.: (Not) yet another matcher. In: CIKM. (2009) 1537–1540
14. Lin, D.: An information-theoretic definition of similarity. In: ICML. (1998) 296–304
15. Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA, ACM (2008) 327–336
16. Aumueller, D., Do, H.H., Massmann, S., Rahm, E.: Schema and ontology matching with COMA++. In: ACM SIGMOD. (2005) 906–908
17. Yatskevich, M.: Preliminary evaluation of schema matching systems. Technical Report DIT-03-028, Informatica e Telecomunicazioni, University of Trento (2003)
18. Diestel, R.: Graph Theory (Graduate Texts in Mathematics). Springer (August 2005)